

OSLab: An Interactive Operating System Laboratory

by Markus Wulff and Torsten Braun

The Operating System Laboratory (OSLab) project promoted by the Swiss Virtual Campus (SVC) is an online course that will teach the learner about the principles of computer operating systems using a progressive approach and problem-oriented learning. OSLab focuses on a hands-on training experience for the learner.

Operating systems provide the basic functionality on every computer that allows any type of application software to be run. For computer science students it is crucial to know and understand the principal concepts and mechanisms of operating systems. This understanding helps them to efficiently use programming languages to develop software built on top of the operating system. This also applies to people working in other fields related to information technology, who could benefit greatly from a deeper knowledge of operating system internals.

Instead of teaching operating systems exclusively in traditional classroom lectures, OSLab provides a modular e-learning course with a strong focus on hands-on training. The course currently consists of seven modules that deal with some of the most elementary topics of operating system architecture: process scheduling, inter-process communication, memory management, file systems, distributed file systems, security, device drivers and I/O (input/output). The modular concept of the course allows it to be extended incrementally as needed.

Each course module is divided into four chapters and has the structure shown in Figure 1. Different tools are used to keep track of the learners' progress. Chapter 1 contains an introduction to and information on the goals and content of the module. In Chapter 2 the theoretical part of the module is presented, and it concludes with an intermediate quiz. Chapter 3 contains the necessary instructions for hands-on exercises, including links to the appropriate training computers. The module ends with Chapter 4, in which the learner must solve a final quiz and give general feedback about the experience of completing the module.

Traditional and advanced learning tools and methods are blended to achieve a

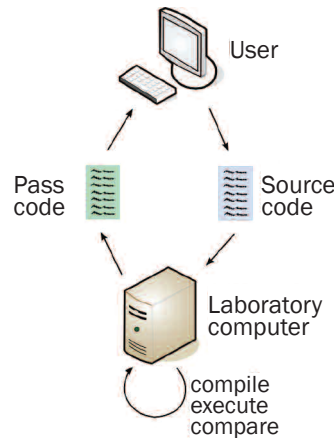


Figure 1: Structure of an OSLab course module; every module has the same structure and appearance.

better learning performance. Traditional tools are known from offline courses, while advanced tools take advantage of the new possibilities offered by an online course infrastructure.

The traditional tools consist of the self test, quiz and final quiz. The self test allows learners to determine their knowledge state without being evaluated by the tutor. In the quiz, learners must demonstrate their mastery of the

knowledge acquired in the theory chapter. With the final quiz, learners summarize the results obtained by performing the practical tasks from the hands-on chapter. At each stage, the Interactive Theory window may be used to provide remote access to a Linux computer so that commands and features described in the learning material can be tried out.

The advanced tools consist of a discussion board, a logbook, and special sections like 'My goals', personal synthesis and survey. The discussion forum helps learners exchange knowledge and obtain support from other learners or the tutor(s). The logbook allows learners to keep track of the learning progress and to write down notes for future reference. The discussion and logbook are available throughout the whole module. The 'My Goals' section requires learners to formulate their learning goals based on the introduction, and thereby activates prior knowledge. The personal synthesis helps learners to structure acquired knowledge from the theory and hands-on chapters. It also allows comparison of the initially specified and finally achieved goals. In the survey users are

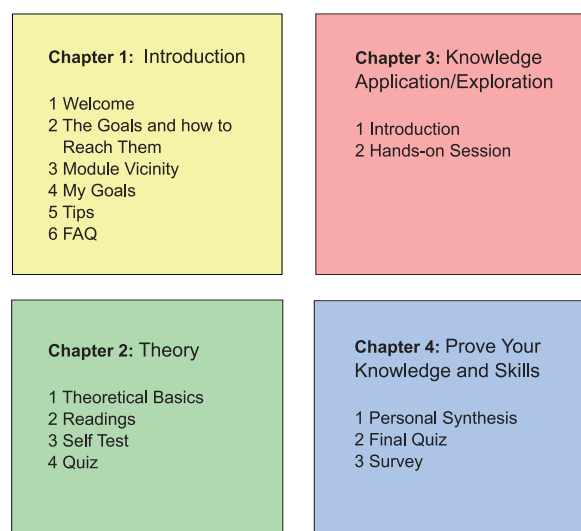


Figure 2: The hands-on part of the module requires programming exercises to be solved. Users send the source code of the solution to the laboratory computer where it is compiled and executed. The output of the learners' program produced from the test input data is compared to the output of a reference program. In return, the user gets a pass code containing all the information necessary to evaluate the solution.

asked for feedback to help the tutor and course developers to adapt and improve the learning module.

In addition to the learning modules, a Java applet has been developed in order to present a common user interface for the hands-on exercises. The applet can be adapted to the particular requirements of any given exercise. Learners use this interface to send the solution for the given task to the server, and in return receive a 'pass code' as well as possible log/error messages. At the back-end on the training computer in the laboratory, a compiler and/or simulator is running to compare the presented solution with a given reference solution using the same input data for both. The pass code is delivered to the course tutor, who uses a complementary tool to decode it and review the solution. Figure 2 illustrates this procedure.

This e-learning course will complement traditional lectures, especially in topics for which hands-on training is feasible and a higher rate of learning success and sustainability can be expected. The course and learning modules, which are in themselves closed, allow tutors to choose some or all of the selection of modules and to add new modules in later phases.

The course is being developed and maintained by several universities in Switzerland. The theory pages, which along with the hands-on exercises are currently hosted on a WebCT learning management system, are connected to a distributed authentication and authori-

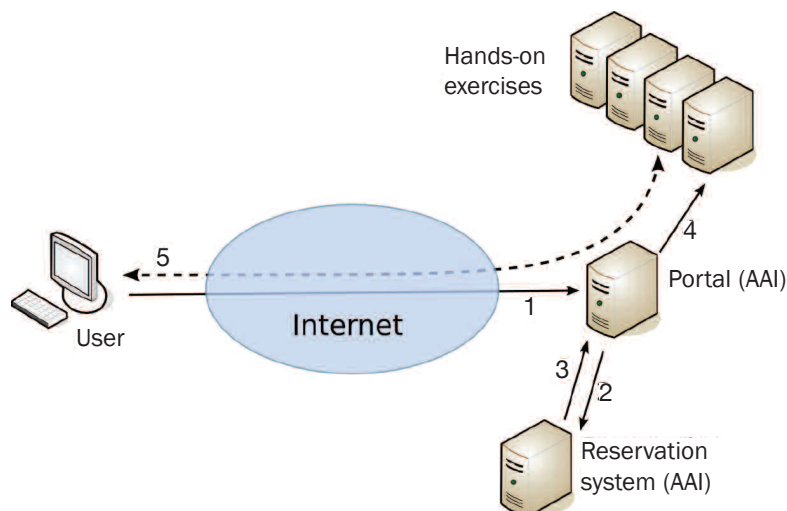


Figure 3: Architecture of the hands-on infrastructure. The laboratory machines are protected by a portal computer. Users send a request for a session to the portal (1). Due to limited resources, a reservation system is used to manage access to the exercise computers (2,3). Only if a free timeslot is available is access to the respective module granted (4) and the session established (5).

zation infrastructure (AAI) and can also be made available to learners from outside the project partner universities. The AAI is coordinated by SWITCH, the Swiss research and education network, and provides single sign-on Web access for all Swiss universities. The architecture of the hands-on infrastructure is shown in Figure 3.

With the cooperation of several institutions, it is possible to share both resources and knowledge. Each partner operates and shares its hands-on training laboratories. Learners profit from a didactically structured and graphically appealing course with interesting interactive and animated learning content and the possibility to individually perform hands-on training around the clock.

Two modules of the OSLab course were successfully tested during the operating systems lecture at the University of Bern in the summer semester of 2007. In the remaining project time until mid-2008, the modules will be completed and tested by students of the partner universities.

Links:

<http://www.oslab.ch>
<http://www.switch.ch/aai/>
<http://www.virtualcampus.ch>
<http://www.webct.com>

Please contact:

Markus Wulff
 University of Bern, Switzerland
 Tel: +41 31 631 86 47
 E-mail: mwulff@iam.unibe.ch